

Muhammad Affandes

Materi Praktikum

Algoritma & Pemrograman

Menggunakan

JAVA

Muhammad Affandes

BAGIAN 1

METHOD

1.1 Method Sederhana

#1 Membuat method sederhana

```
public class Method01 {  
    public static void main(String[] args) {  
        salam();  
    }  
    public static void salam() {  
        System.out.println("Assalammu'alaikum");  
    }  
}
```

#2 Membuat method sederhana dengan argument

```
public class Method02 {  
    public static void main(String[] args) {  
        salam("Affandes");  
    }  
    public static void salam(String nama) {  
        System.out.printf("Assalammu'alaikum " +  
nama);  
    }  
}
```

#3 Urutan method tidak berpengaruh

```
public class Method03 {  
    public static void salam(String nama, int  
jumlah) {  
        for (int i = 0; i < jumlah; i++) {  
            System.out.println("Assalammu'alaikum "  
+ nama + ", ");  
        }  
    }  
    public static void main(String[] args) {  
        salam("Affandes",5);  
    }  
}
```

1.2 Argument pada Method

#4 Argument lebih dari satu

```
public class Method04 {  
    public static void main(String[] args) {  
        jumlah(9,2);  
    }  
    public static void jumlah(int a, int b) {  
        System.out.println("Jumlah = " + (a+b));  
    }  
}
```

#5 Argument dengan tipe data yang berbeda

```
public class Method05 {  
    public static void cetak(String nama, boolean  
status) {  
        System.out.print(nama + );  
        if( status ) {  
            System.out.println(" (Menikah)");  
        } else {  
            System.out.println(" (Belum Menikah)");  
        }  
    }  
}
```

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
public static void main(String[] args) {  
    cetak("Affandes", true);  
}
```

#6 Argument dengan tipe data yang berbeda

```
public class Method06 {  
    public static void cetak(char x, int y) {  
        for (int i = 0; i < y; i++) {  
            for (int j = 0; j < y; j++) {  
                System.out.print(x);  
            }  
            System.out.println();  
        }  
    }  
    public static void main(String[] args) {  
        cetak('*', 4);  
    }  
}
```

#7 Variabel sebagai argument

```
public class Method07 {  
    public static void main(String[] args) {  
        String nama = "Affandes";  
        int umur = 21;  
        cetak(nama, umur);  
    }  
    public static void cetak(String nama, int umur)  
    {  
        System.out.println(nama + " (" + umur + "  
tahun)");  
    }  
}
```

#8 Method dengan tiga argument

```
public class Method08 {  
    public static void cetak(String nim, String  
nama, double nilai) {  
        System.out.println("Biodata Mahasiswa");  
        System.out.println("=====");  
        System.out.println("Nama \t: " + nama);  
        System.out.println("NIM \t: " + nim);  
        System.out.printf("Nilai \t: %3.2f \n",  
nilai);  
        System.out.println("=====");  
    }  
    public static void main(String[] args) {  
        cetak("123", "Affandes", 87.9);  
        cetak("124", "Budi", 86.5);  
    }  
}
```

1.3 Void

#9 Method void

```
public class Method09 {  
    public static void jumlahDenganVoid(int a, int  
b) {  
        int jumlah = a + b;  
        System.out.println("Jumlah = " + jumlah);  
    }  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 13;  
  
        jumlahDenganVoid(a, b);  
    }  
}
```

1.4 Non-Void

#10 Method Non Void 1

```
public class Method10 {  
    public static int jumlahDenganNonVoid(int a,  
int b) {  
        int jumlah = a + b;  
        return jumlah;  
    }  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 13;  
  
        int jumlah = jumlahDenganNonVoid(a, b);  
        System.out.println("Jumlah = " + jumlah);  
    }  
}
```

#11 Method Non Void 2

```
public class Method11 {  
    public static double  
hitungLuasPersegiPanjang(double panjang, double  
lebar) {  
        double luas = panjang * lebar;  
        return luas;  
    }  
    public static double input(String label) {  
        Scanner sc = new Scanner(System.in);  
        System.out.print("Input " + label + ": ");  
        return sc.nextDouble();  
    }  
    public static void main(String[] args) {  
        double panjang = input("Panjang");  
        double lebar = input("Lebar");  
        System.out.println("Luas = " +  
hitungLuasPersegiPanjang(panjang, lebar));  
    }  
}
```

#12 Method Non Void 3

```
public class Method12 {  
    public static int  
hitungLuasPermukaanDadu(double panjangSisi)  
    {  
        return 6 * panjangSisi * panjangSisi;  
    }  
    public static void main(String[] args) {  
        Scanner sc = new Scanner(System.in);  
        System.out.println("Aplikasi Luas Dadu");  
        System.out.println("=====");  
        System.out.print("Panjang Sisi: ");  
        double panjangSisi = sc.nextDouble();  
        System.out.println("-----");  
        double luas =  
hitungLuasPermukaanDadu(panjangSisi);  
        System.out.println("Luas: " + luas);  
    }  
}
```

#13 Method Non Void 4

```
public class Method13 {  
    public static int jumlah(int a, int b) {  
        int c = a + b;  
        System.out.println("c = " + c);  
    }  
    public static void main(String[] args) {  
        jumlah(6,2);  
    }  
}
```

1.5 Main

#14 Method Main dengan huruf M kapital

```
public class Method14 {  
    public static void Main(String[] args) {  
        System.out.println("Salam");  
    }  
}
```

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
}  
}
```

#15 Method main tanpa argument

```
public class Method15 {  
    public static void main() {  
        System.out.println("Salam");  
    }  
}
```

#16 Method main tanpa static modifier

```
public class Method16 {  
    public void main(String[] args) {  
        System.out.println("Salam");  
    }  
}
```

#17 Method main tanpa public modifier

```
public class Method17 {  
    static void main(String[] args) {  
        System.out.println("Salam");  
    }  
}
```

#18 Method main tanpa void modifier

```
public class Method18 {  
    public static main(String[] args) {  
        System.out.println("Salam");  
    }  
}
```

#19 Method main dengan argument berbeda 1

```
public class Method19 {  
    public static void main(String[] args, int x) {  
        System.out.println("Salam");  
    }  
}
```


Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
}  
}
```

#20 Method main dengan argument berbeda 2

```
public class Method20 {  
    public static void main(String args) {  
        System.out.println("Salam");  
    }  
}
```

BAGIAN 2

CLASS DAN OBJECT

2.1 Class

#1 Membuat class Persegi

```
public class Persegi {  
    // ini adalah sebuah class  
    // dengan nama 'Persegi'  
}
```

2.2 Object

#2 Membuat object class Persegi dari class lain

```
public class MembuatObject {  
    public static void main(String[] args) {  
        Persegi x = new Persegi();  
    }  
}
```

#3 Membuat object class Persegi dari class Persegi

```
public class Persegi {  
    public static void main(String[] args) {  
        Persegi x = new Persegi();  
    }  
}
```

#4 Membuat object dan memanggil methodnya

```
public class Persegi {  
    public static void main(String[] args) {  
        Persegi x = new Persegi();  
        x.luas(6,5);  
    }  
    public void luas(int p, int l) {  
        System.out.println("Luas = " + (p * l));  
    }  
}
```

#5 Membuat object dan memanggil methodnya dari class lain

```
public class MembuatObject {  
    public static void main(String[] args) {  
        Persegi x = new Persegi();  
        x.luas(8,5);  
    }  
}
```

2.3 Constructor

#6 Membuat dan menggunakan constructor 1

```
public class PersegiPanjang {  
    // Constructor  
    public PersegiPanjang(int p, int l) {  
        int luas = p * l;  
        System.out.println("Luas = " + luas);  
    }  
    // Method main  
    public static void main(String[] args) {  
        PersegiPanjang x = new PersegiPanjang(8,  
5);  
    }  
}
```

#7 Membuat dan menggunakan constructor 2

```
public class PersegiPanjang {  
    // Constructor  
    public PersegiPanjang(int p, int l) {  
        int luas = p * l;  
        System.out.println("Luas = " + luas);  
    }  
    // Method luas  
    public void luas(int p, int l) {  
        int luas = p * l;  
        System.out.println("Luas = " + luas);  
    }  
    // Method main  
    public static void main(String[] args) {  
        PersegiPanjang x = new PersegiPanjang(8,  
5);  
        x.luas(6,4);  
    }  
}
```

2.4 Static

#8 Method static dan method non-static

```
public class Kubus {  
    public static void jumlahStatic(int a, int b) {  
        int jumlah = a + b;  
        System.out.println("Jumlah = " + jumlah);  
    }  
    public void jumlahNonStatic(int a, int b) {  
        int jumlah = a + b;  
        System.out.println("Jumlah = " + jumlah);  
    }  
    public static void main(String[] args) {  
        // Panggil method static  
        Kubus.jumlahStatic(10,9);  
        // Panggil method non-static  
        Kubus x = new Kubus();  
        x.jumlahNonStatic(23,7);  
    }  
}
```

```
}  
}
```

Memanggil method dari class lain

```
public class MemanggilKubus {  
    public static void main(String[] args) {  
        // Panggil method static  
        Kubus.jumlahStatic(8,5);  
        // Panggil method non-static  
        Kubus x = new Kubus();  
        x.jumlahNonStatic(7,5);  
    }  
}
```

2.5 Non-Static

Method non-static

```
public class Lingkaran {  
    public double luasPakeJariJari(double jariJari)  
    {  
        return Math.PI * Math.pow(jariJari, 2);  
    }  
    public double luasPakeDiameter(double diameter)  
    {  
        return Math.PI * Math.pow(diameter, 2) / 4;  
    }  
    public double kelilingPakeJariJari(double jariJari) {  
        return Math.PI * jariJari * 2;  
    }  
    public double kelilingPakeDiameter(double diameter) {  
        return Math.PI * diameter;  
    }  
    public static void main(String[] args) {  
        Lingkaran x = new Lingkaran();  
        double jariJari = 7;  
        double diameter = 14;  
    }  
}
```

```
        System.out.println("Luas (r): " +
x.luasPakeJariJari(jariJari));
        System.out.println("Luas (d): " +
x.luasPakeDiameter(diameter));
        System.out.println("Keliling (r): " +
x.kelilingPakeJariJari(jariJari));
        System.out.println("Keliling (d): " +
x.kelilingPakeDiameter(diameter));
    }
}
```

Memanggil method non-static dari class lain

```
public class MemanggilLingkaran {
    public static void main(String[] args) {
        Lingkaran x = new Lingkaran();
        double jariJari = 7;
        double diameter = 14;

        System.out.println("Luas (r): " +
x.luasPakeJariJari(jariJari));
        System.out.println("Luas (d): " +
x.luasPakeDiameter(diameter));
        System.out.println("Keliling (r): " +
x.kelilingPakeJariJari(jariJari));
        System.out.println("Keliling (d): " +
x.kelilingPakeDiameter(diameter));
    }
}
```

2.6 Attribute

Attribute pada class Persegi

```
public class Persegi {
    public double panjang, lebar;

    public static void main(String[] args) {
        Persegi obj = new Persegi();
    }
}
```

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
        obj.panjang = 4;
        obj.lebar = 3;

        System.out.println("Luas = " + (obj.panjang
* obj.lebar));
    }
}
```

Menggunakan attribute pada method

```
public class Persegi {
    public double panjang, lebar;

    public double luas() {
        return panjang * lebar;
    }

    public static void main(String[] args) {
        Persegi obj = new Persegi();
        obj.panjang = 4;
        obj.lebar = 3;

        System.out.println("Luas = " + obj.luas());
    }
}
```

Menggunakan attribute pada constructor

```
public class Persegi {
    public double panjang, lebar;

    public double luas() {
        return panjang * lebar;
    }

    public Persegi(int p, int l) {
        panjang = p;
        lebar = l;
    }

    public static void main(String[] args) {
        Persegi obj = new Persegi(4,3);
        System.out.println("Luas = " + obj.luas());
    }
}
```

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
}  
}
```

Memanggil method yang menggunakan attribute dari class lain

```
public class AppPersegi {  
    public static void main(String[] args) {  
        Persegi obj = new Persegi(4,3);  
        System.out.println("Luas = " + obj.luas());  
    }  
}
```


BAGIAN 3

ARRAY

3.1 Array Sederhana

Array sederhana 1

```
public class ArraySederhana {  
    public static void main(String[] args) {  
        String[] x = {"Ali", "Budi", "Citra"};  
    }  
}
```

Array sederhana 2

```
public class ArraySederhana {  
    public static void main(String[] args) {  
        String[] x = new String[3];  
        x[0] = "Ali";  
        x[1] = "Budi";  
        x[2] = "Citra";  
    }  
}
```

3.2 Membuat Array Sederhana

Membuat Array sederhana 1

```
public class ArraySederhana {  
    public static void main(String[] args) {  
        int[] x = {5,1,6,7};  
    }  
}
```

Membuat Array sederhana 2

```
public class ArraySederhana {
    public static void main(String[] args) {
        int[] x = new int[4];
        x[0] = 5;
        x[1] = 1;
        x[2] = 6;
        x[3] = 7;
    }
}
```

Variasi membuat Array

```
public class ArraySederhana {
    public static void main(String[] args) {
        int[] x = new int[4];
        int y[] = new int[4];

        x[0] = 5;
        x[1] = 1;
        x[2] = 6;
        x[3] = 7;

        y[0] = 5;
        y[1] = 1;
        y[2] = 6;
        y[3] = 7;
    }
}
```

3.3 Menggunakan Array Sederhana

Menggunakan Array 1

```
public class MenggunakanArray {
    public static void main(String[] args) {
        // Membuat Array
        int[] x = new int[4];
        x[0] = 5;
```

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
        x[1] = 1;
        x[2] = 6;
        x[3] = 7;

        // Menggunakan Array
        System.out.println("X indeks ke 0 = " +
x[0]);
        System.out.println("X indeks ke 1 = " +
x[1]);
        System.out.println("X indeks ke 2 = " +
x[2]);
        System.out.println("X indeks ke 3 = " +
x[3]);
    }
}
```

Menggunakan Array 2

```
public class MenggunakanArray {
    public static void main(String[] args) {
        // Membuat Array
        int[] x = new int[4];
        x[0] = 5;
        x[1] = 1;
        x[2] = 6;
        x[3] = 7;

        // Menggunakan Array
        for (int i = 0; i < x.length; i++) {
            System.out.println("X indeks ke " + i +
" = " + x[i]);
        }
    }
}
```

Menggunakan Array 3

```
public class MenggunakanArray2 {
    public static void main(String[] args) {
        for (int i = 0; i < args.length; i++) {
            System.out.println("Argument " + i + "
= " + args[i]);
        }
    }
}
```

```
}  
}
```

3.4 Mengubah Array Sederhana

Mengubah isi Array

```
public class MengubahArray {  
    public static void main(String[] args) {  
        String[] nama =  
        {"Ali", "Budi", "Citra", "Dodi"};  
  
        nama[1] = "Bagas";  
  
        for (int i = 0; i < nama.length; i++) {  
            System.out.println("nama[" + i + "] = " +  
nama[i]);  
        }  
    }  
}
```

Mengubah isi Array

```
public class MengubahArray {  
    public static void main(String[] args) {  
        String[] nama =  
        {"Ali", "Budi", "Citra", "Dodi"};  
  
        for (int i = 0; i < nama.length; i++) {  
            if(i == 2) {  
                nama[i] = "Bagas";  
            }  
        }  
    }  
}
```

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
        for (int i = 0; i < nama.length; i++) {
            System.out.println("nama["+i+"] = " +
nama[i]);
        }
    }
}
```

Array sebagai parameter pada method

```
public class ArrayParam {
    public static void jumlah(int[] data) {
        int total = 0;
        for (int i = 0; i < data.length; i++) {
            total += data[i];
        }
        System.out.println("Total = " + total);
    }

    public static void main(String[] args) {
        int x[] = {5,2,5,5,3};
        jumlah(x);
    }
}
```

Array sebagai attribute pada method

```
public class AttributeArray {
    public int[] data = new int[4];

    public void jumlah() {
        int total = 0;
        for (int i = 0; i < data.length; i++) {
            total += data[i];
        }
        System.out.println("Total = " + total);
    }

    public static void main(String[] args) {
        AttributeArray obj = new AttributeArray();
        obj.data[0] = 5;
        obj.data[1] = 1;
        obj.data[2] = 2;
        obj.data[3] = 6;
    }
}
```

```
        obj.jumlah();  
    }  
}
```

Array sebagai attribute pada method

```
public class AttributeArray {  
    public int[] data;  
  
    public void jumlah() {  
        int total = 0;  
        for (int i = 0; i < data.length; i++) {  
            total += data[i];  
        }  
        System.out.println("Total = " + total);  
    }  
  
    public static void main(String[] args) {  
        int x[] = {5,6,1,5};  
        AttributeArray obj = new AttributeArray();  
        obj.data = x;  
        obj.jumlah();  
    }  
}
```

3.5 Array Multidimensi

Contoh Array multidimensi

```
public class ArrayMulti {  
    public static void main(String[] args) {  
        int[][] x = {{1,2,3},{4,5,6},{7,8,9}};  
        for (int i = 0; i < x.length; i++) {  
            for (int j = 0; j < x[i].length; j++) {  
                System.out.print(x[i][j] + " ");  
            }  
            System.out.println();  
        }  
    }  
}
```

3.6 Membuat Array Multidimensi

Membuat Array Multidimensi 1

```
public class MembuatArrayMulti {  
    public static void main(String[] args) {  
        int[][] x = {{1,2},{3,4}};  
        int[] y[] = {{1,2},{3,4}};  
        int z[][] = {{1,2},{3,4}};  
    }  
}
```

Membuat Array Multidimensi 2

```
public class MembuatArrayMulti {  
    public static void main(String[] args) {  
        int[][] x = new int[3][2];  
        x[0][0] = 4;  
        x[0][1] = 6;  
        x[1][0] = 1;  
        x[1][1] = 6;  
        x[2][0] = 7;  
        x[2][1] = 8;  
    }  
}
```

3.7 Menggunakan Array Multidimensi

Menggunakan Array Multidimensi

```
public class MembuatArrayMulti {  
    public static void main(String[] args) {  
        int[][] x = new int[3][2];  
        x[0][0] = 4;  
        x[0][1] = 6;  
        x[1][0] = 1;  
        x[1][1] = 6;  
        x[2][0] = 7;  
        x[2][1] = 8;  
    }  
}
```

```
        for (int i = 0; i < x.length; i++) {
            for (int j = 0; j < x[i].length; j++) {
                System.out.print(x[i][j] + " ");
            }
            System.out.println();
        }
    }
}
```

Menggunakan Arrah Multidimensi pada method

```
public class MethodArrayMulti {
    public static void main(String[] args) {
        String[][] mahasiswa = {
            {"10020012", "Andrian Hidayat"},
            {"10020014", "Bestari Wijaya"},
            {"10020015", "Chandra Putra"},
        };

        MethodArrayMulti obj = new
MethodArrayMulti();
        obj.cetak(mahasiswa);
    }

    public void cetak(String[][] daftarMhs) {
        for (String[] mhs : daftarMhs) {
            for (String kolom : mhs) {
                System.out.print(kolom + " ");
            }
            System.out.println();
        }
    }
}
```


BAGIAN 4

SEARCHING

4.1 Sequential Search

Pencarian sederhana

```
public class SearchingDasar {  
    public static void main(String[] args) {  
        int[] data = {5,12,23,31,6,12,37};  
        int cari = 6;  
        for (int i = 0; i < data.length; i++) {  
            if( data[i] == cari ) {  
                System.out.println("Data " + cari +  
" ditemukan di indeks " + i);  
            }  
        }  
    }  
}
```

Pencarian dengan hasil lebih dari satu

```
public class SearchingDasar {  
    public static void main(String[] args) {  
        int[] data = {5,12,23,31,6,12,37};  
        int cari = 12;  
        for (int i = 0; i < data.length; i++) {  
            if( data[i] == cari ) {  
                System.out.println("Data " + cari +  
" ditemukan di indeks " + i);  
            }  
        }  
    }  
}
```

Pencarian dengan hasil hanya satu

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
public class SearchingDasar {
    public static void main(String[] args) {
        int[] data = {5,12,23,31,6,12,37};
        int cari = 12;
        for (int i = 0; i < data.length; i++) {
            if( data[i] == cari ) {
                System.out.println("Data " + cari +
" ditemukan di indeks " + i);
                break;
            }
        }
    }
}
```

Pencarian dengan hasil hanya satu dan dimulai dari belakang

```
public class SearchingDasar {
    public static void main(String[] args) {
        int[] data = {5,12,23,31,6,12,37};
        int cari = 12;
        for (int i = data.length-1; i >= 0; i--) {
            if( data[i] == cari ) {
                System.out.println("Data " + cari +
" ditemukan di indeks " + i);
                break;
            }
        }
    }
}
```

Pencarian dengan status tidak ditemukan

```
public class SearchingDasar {
    public static void main(String[] args) {
        int[] data = {5,12,23,31,6,12,37};
        int cari = 99;
        boolean status = false;
        int index = 0;
        for (int i = data.length-1; i >= 0; i--) {
            if( data[i] == cari ) {
                status = true;
            }
        }
    }
}
```

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
        index = i;
    }
}
if(status) {
    System.out.println("Data " + cari +
ditemukan di indeks " + index);
} else {
    System.out.println("Data tidak dapat
ditemukan.");
}
}
}
```

Pencarian menggunakan method static

```
public class SearchingDasar {
    public static void main(String[] args) {
        int[] data = {5,12,23,31,6,12,37};
        int cari = 99;
        int hasil = cariData(data, cari);
        if(hasil >= 0) {
            System.out.println("Data " + cari +
ditemukan di index " + hasil);
        } else {
            System.out.println("Data tidak dapat
ditemukan.");
        }
    }
    public static int cariData(int[] data, int
cari) {
        int index = -1;
        for (int i = data.length-1; i >= 0; i--) {
            if( data[i] == cari ) {
                index = i;
            }
        }
        return index;
    }
}
```

Pencarian menggunakan method non-static

```
public class SearchingDasar {
    public static void main(String[] args) {
        int[] data = {5,12,23,31,6,12,37};
        int cari = 99;
        SearchingDasar obj = new SearchingDasar();
        int hasil = obj.cariData(data, cari);
        if(hasil >= 0) {
            System.out.println("Data " + cari +
ditemukan di index " + hasil);
        } else {
            System.out.println("Data tidak dapat
ditemukan.");
        }
    }
    public int cariData(int[] data, int cari) {
        int index = -1;
        for (int i = data.length-1; i >= 0; i--) {
            if( data[i] == cari ) {
                index = i;
            }
        }
        return index;
    }
}
```

Pencarian menggunakan class dan object

```
public class Sequential {
    private int[] data;

    public int[] getData() {
        return data;
    }

    public void setData(int[] data) {
        this.data = data;
    }

    public int cari(int value) {
        int index = -1;
        for (int i = 0; i < data.length; i++) {
```

```
        if(data[i] == value) {  
            index = i;  
            break;  
        }  
    }  
    return index;  
}  
}
```

4.2 Binary Search

Binary Search Sederhana

```
public class BinarySearch {  
    public static void main(String[] args) {  
        int[] data =  
{12,15,17,19,21,24,28,32,36,37,37,63};  
        int cari = 28;  
        int awal = 0;  
        int akhir = data.length-1;  
        while (awal < akhir) {  
            int tengah = (awal + akhir) / 2;  
            if( data[tengah] == cari ) {  
                System.out.println("Data " + cari +  
" ada di indeks " + tengah);  
                break;  
            } else if( cari < data[tengah] ) {  
                akhir = tengah;  
            } else {  
                awal = tengah;  
            }  
        }  
        for (int i = 0; i < data.length; i++) {  
            System.out.print(data[i] + " ");  
        }  
    }  
}
```

Binary Search dalam bentuk class

```
public class BinarySearchClass {
    private int data[];

    public BinarySearchClass(int data[]) {
        this.data = data;
    }

    public int cari(int v) {
        return cari(v, 0, data.length-1);
    }

    public int cari(int v, int awal, int akhir) {
        int tengah = (awal + akhir) / 2;
        if( awal > akhir ) {
            return -1;
        } else if( data[tengah] == v ) {
            return tengah;
        } else if( v < data[tengah] ) {
            return cari(v, awal, tengah);
        } else {
            return cari(v, tengah, akhir);
        }
    }
}

public class AppBinary {
    public static void main(String[] args) {
        int[] data =
{12,15,17,19,21,24,28,32,36,37,37,63};
        BinarySearchClass obj = new
BinarySearchClass(data);
        int hasil = obj.cari(28);
        if( hasil < 0 ) {
            System.out.println("Data 28 tidak
ditemukan");
        } else {
            System.out.println("Data 28 ditemukan
di indeks " + hasil);
        }
    }
}
```

BAGIAN 5

SORTING

5.1 Bubble Sort

Bubble Sort Dasar

```
public class BubbleSortDasar {
    public static void main(String[] args) {
        int data[] =
{32,56,21,43,7,18,38,41,22,27};
        for (int i = 0; i < data.length-1; i++) {
            for (int j = i+1; j < data.length; j++)
            {
                if(data[i] > data[j]) {
                    int temp = data[i];
                    data[i] = data[j];
                    data[j] = temp;
                }
            }
            for (int i = 0; i < data.length; i++) {
                System.out.print(data[i] + " ");
            }
        }
    }
}
```

Bubble Sort dengan method cetak

```
public class BubbleSortDasar {
    public static void main(String[] args) {
        int data[] =
{32,56,21,43,7,18,38,41,22,27};
        for (int i = 0; i < data.length-1; i++) {
            for (int j = i+1; j < data.length; j++)
            {

```

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
        if(data[i] > data[j]) {
            int temp = data[i];
            data[i] = data[j];
            data[j] = temp;
        }
    }
    cetak(data);
}

public static void cetak(int data[]) {
    for (int i = 0; i < data.length; i++) {
        System.out.print(data[i] + " ");
    }
}
}
```

Bubble Sort dengan bentuk Class

```
public class BubbleSortClass {
    private int[] data;

    public BubbleSortClass(int[] data) {
        this.data = data;
    }

    public void cetak() {
        for (int d : data) {
            System.out.print(d + " ");
        }
        System.out.println();
    }

    public void sort() {
        for (int i = 0; i < data.length-1; i++) {
            for (int j = i+1; j < data.length; j++)
            {
                if(data[i] > data[j]) {
                    int temp = data[i];
                    data[i] = data[j];
                    data[j] = temp;
                }
            }
        }
    }
}
```


Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
    }  
}  
  
public class AppBubble {  
    public static void main(String[] args) {  
        int[] data =  
{24,12,32,63,19,21,28,37,15,17,37,36};  
        BubbleSortClass obj = new  
BubbleSortClass(data);  
        obj.cetak();  
        obj.sort();  
        obj.cetak();  
    }  
}
```

Bubble Sort dengan method Ascending dan Descending

```
public class BubbleSortClass {  
    private int[] data;  
  
    public BubbleSortClass(int[] data) {  
        this.data = data;  
    }  
  
    public void cetak() {  
        for (int d : data) {  
            System.out.print(d + " ");  
        }  
        System.out.println();  
    }  
  
    public void sort() {  
        for (int i = 0; i < data.length-1; i++) {  
            for (int j = i+1; j < data.length; j++)  
{  
                if(data[i] > data[j]) {  
                    int temp = data[i];  
                    data[i] = data[j];  
                    data[j] = temp;  
                }  
            }  
        }  
    }  
}
```

```
    }
}

public void sortDesc() {
    for (int i = 0; i < data.length-1; i++) {
        for (int j = i+1; j < data.length; j++)
        {
            if(data[i] < data[j]) {
                int temp = data[i];
                data[i] = data[j];
                data[j] = temp;
            }
        }
    }
}

public class AppBubble {
    public static void main(String[] args) {
        int[] data =
{24,12,32,63,19,21,28,37,15,17,37,36};
        BubbleSortClass obj = new
BubbleSortClass(data);
        obj.cetak();
        obj.sort();
        obj.cetak();
        obj.sortDesc();
        obj.cetak();
    }
}
```

5.2 Insertion Sort

Insertion Sort Dasar

```
public class InsertionSortDasar {
    public static void main(String[] args) {
        int[] data =
{52,72,34,11,37,63,35,8,25,22,6};
        for (int i = 1; i < data.length; i++) {
            for (int j = i; j >= 1; j--) {
```

Materi Praktikum

Algoritma dan Pemrograman Menggunakan Java

```
        if(data[j] > data[j-1]) {
            break;
        }
        int temp = data[j];
        data[j] = data[j-1];
        data[j-1] = temp;
    }
}
for (int i = 0; i < data.length; i++) {
    System.out.print(data[i] + " ");
}
}
```

Insertion Sort menggunakan method

```
public class InsertionSortDasar {
    public static void main(String[] args) {
        int[] data =
{52,72,34,11,37,63,35,8,25,22,6};
        for (int i = 1; i < data.length; i++) {
            for (int j = i; j >= 1; j--) {
                if(data[j] > data[j-1]) {
                    break;
                }
                int temp = data[j];
                data[j] = data[j-1];
                data[j-1] = temp;
            }
        }
        cetak(data);
    }
    public static void cetak(int data[]) {
        for (int i = 0; i < data.length; i++) {
            System.out.print(data[i] + " ");
        }
    }
}
```

Insertion Sort dalam bentuk class

```
public class InsertionSortClass {
    private int data[];
```

```
public InsertionSortClass(int data[]) {
    this.data = data;
}

public void cetak() {
    for (int i = 0; i < data.length; i++) {
        System.out.print(data[i] + " ");
    }
    System.out.println();
}

public void sort() {
    for (int i = 1; i < data.length; i++) {
        for (int j = i; j >= 1; j--) {
            if(data[j] > data[j-1]) {
                break;
            }
            tukar(j,j-1);
        }
    }
}

public void sortDesc() {
    for (int i = 1; i < data.length; i++) {
        for (int j = i; j >= 1; j--) {
            if(data[j] < data[j-1]) {
                break;
            }
            tukar(j,j-1);
        }
    }
}

public void tukar(int x, int y) {
    int temp = data[x];
    data[x] = data[y];
    data[y] = temp;
}

}

public class AppInsertion {
    public static void main(String[] args) {
        int[] data =
```

```
{24,12,32,63,19,21,28,37,15,17,37,36};
    InsertionSortClass obj = new
InsertionSortClass(data);
    obj.cetak();
    obj.sort();
    obj.cetak();
    obj.sortDesc();
    obj.cetak();
}
}
```

5.3 Selection Sort

Selection Sort Dasar

```
public class SelectionSortDasar {
    public static void main(String[] args) {
        int data[] =
{32,56,21,43,7,18,38,41,22,27};
        for (int i = 0; i < data.length-1; i++) {
            int max = i;
            for (int j = i; j < data.length; j++) {
                if(data[j] < data[max]) {
                    max = j;
                }
            }
            int temp = data[i];
            data[i] = data[max];
            data[max] = temp;
        }
        for (int i = 0; i < data.length; i++) {
            System.out.print(data[i] + " ");
        }
    }
}
```

Selection Sort dengan method cetak

```
public class SelectionSortDasar {
    public static void main(String[] args) {
        int data[] =
{32,56,21,43,7,18,38,41,22,27};
        for (int i = 0; i < data.length-1; i++) {
            int max = i;
            for (int j = i; j < data.length; j++) {
                if(data[j] < data[max]) {
                    max = j;
                }
            }
            int temp = data[i];
            data[i] = data[max];
            data[max] = temp;
        }
        cetak(data);
    }
    public static void cetak(int data[]) {
        for (int i = 0; i < data.length; i++) {
            System.out.print(data[i] + " ");
        }
    }
}
```

Selection Sort dengan bentuk Class

```
public class SelectionSortClass {
    private int[] data;

    public SelectionSortClass(int[] data) {
        this.data = data;
    }

    public void cetak() {
        for (int d : data) {
            System.out.print(d + " ");
        }
        System.out.println();
    }

    public void sort() {
```

```
        for (int i = 0; i < data.length - 1; i++) {
            int max = i;
            for (int j = i; j < data.length; j++) {
                if (data[j] < data[max]) {
                    max = j;
                }
            }
            int temp = data[i];
            data[i] = data[max];
            data[max] = temp;
        }
    }
}

public class AppSelection {
    public static void main(String[] args) {
        int[] data =
{24,12,32,63,19,21,28,37,15,17,37,36};
        SelectionSortClass obj = new
SelectionSortClass(data);
        obj.cetak();
        obj.sort();
        obj.cetak();
    }
}
```

Selection Sort dengan method Ascending dan Descending

```
public class SelectionSortClass {
    private int[] data;

    public SelectionSortClass(int[] data) {
        this.data = data;
    }

    public void cetak() {
        for (int d : data) {
            System.out.print(d + " ");
        }
        System.out.println();
    }
}
```

```
public void sort() {
    for (int i = 0; i < data.length - 1; i++) {
        int max = i;
        for (int j = i; j < data.length; j++) {
            if(data[j] < data[max]) {
                max = j;
            }
        }
        int temp = data[i];
        data[i] = data[max];
        data[max] = temp;
    }
}

public void sortDesc() {
    for (int i = 0; i < data.length - 1; i++) {
        int max = i;
        for (int j = i; j < data.length; j++) {
            if(data[j] > data[max]) {
                max = j;
            }
        }
        int temp = data[i];
        data[i] = data[max];
        data[max] = temp;
    }
}

}

public class AppSelection {
    public static void main(String[] args) {
        int[] data =
{24,12,32,63,19,21,28,37,15,17,37,36};
        SelectionSortClass obj = new
SelectionSortClass(data);
        obj.cetak();
        obj.sort();
        obj.cetak();
        obj.sortDesc();
        obj.cetak();
    }
}
```


